Technical Disclosure Commons

Defensive Publications Series

July 2020

Determining Viewport Area In User Interfaces With Occluding Elements

Anonymous

Follow this and additional works at: https://www.tdcommons.org/dpubs_series

Recommended Citation
Anonymous, "Determining Viewport Area In User Interfaces With Occluding Elements", Technical Disclosure Commons, (July 02, 2020)
https://www.tdcommons.org/dpubs_series/3399

This work is licensed under a Creative Commons Attribution 4.0 License.
This Article is brought to you for free and open access by Technical Disclosure Commons. It has been accepted for inclusion in Defensive Publications Series by an authorized administrator of Technical Disclosure Commons.

Determining Viewport Area In User Interfaces With Occluding Elements

ABSTRACT

This disclosure describes techniques for the determination of occluded and unoccluded spaces within a user interface. Per techniques of this disclosure, a viewport of a user interface is represented as a set of rectangles determined by recursive subdivision. The area of the viewport is based on a sum of the areas of each unoccluded rectangle in the set of rectangles. The efficient representation of the viewport with a small number of rectangles enables frequent recalculation of the viewport, e.g., even when the viewport is updated frequently in response to user actions such as scrolling.

KEYWORDS

● Viewport
● Display area
● Coordinate window
● Area of interest
● Pop-up window
● Display advertisement
● Ad element
● Online advertising

BACKGROUND

Displayed user interfaces on devices such as computers, tablets, smartphones, etc. include viewports that include content for viewing by users. User interfaces commonly include complex and dynamically changing user interface elements. User interface (UI) elements are often layered, and are sometimes rendered such that individual UI elements are partially or completely arranged over other UI element(s), thereby causing occlusion where one user interface element partially or completely covers another element.

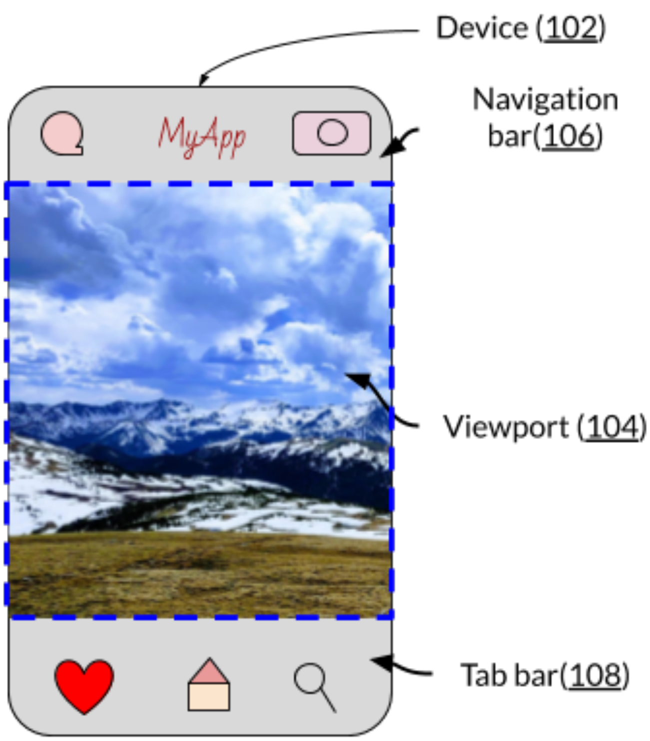

Fig. 1: Viewport of an user interface

Fig. 1 depicts an example user interface on a computing device (102) with a visible content area, or viewport (104) which is the portion of the user interface within which an element of interest is considered visible and is not occluded by other user interface elements. In this illustrative example, the user interface includes occluding elements - a navigation bar (106) and a tab bar (108) that block the main content of the user interface (the image).

Determination of the extent of occlusion and visibility of user interface elements is an important problem, particularly in the case where there may be multiple occluding elements. For example, publishers of online content can include advertisements within the user interface, alongside the content. The publishers bill advertisers based on the visibility of an advertisement that is included within the user interface. Accurate billing can require accurate, robust, and efficient measurement of content area that is actually visible to a user (is not occluded by other user interface elements).

DESCRIPTION

This disclosure describes techniques for the determination of areas of occluded and unoccluded space within a user interface. Per techniques of this disclosure, the viewport of a user interface is obtained based on a set of rectangles. The set of rectangles of a user interface is determined by a recursive subdivision of the user interface. The area of the viewport is the sum of the areas of each unoccluded rectangle in the set of rectangles (the viewport is determined based on unoccluded portions of the user interface).

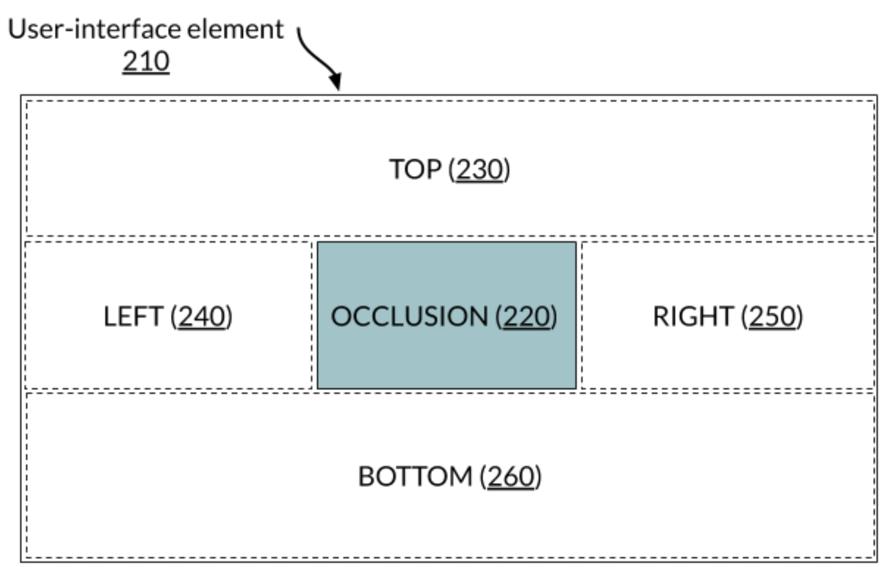

Fig. 2: Unoccluded area is represented with at most 4 rectangles

Fig. 2 depicts an example rectangular user interface element (210) that is occluded by another rectangular user interface element (220). Per techniques of this disclosure, the unoccluded area is represented by 4 rectangles, a top rectangle (230), a bottom rectangle (260), a left rectangle (240), and a right rectangle (250). Determination of the viewport area, the area of the user interface that is visible to a viewer, is made by calculating the total area of the unoccluded rectangles. In this illustrative example, the viewport area is the sum of the areas of the top, bottom, left, and right rectangles.

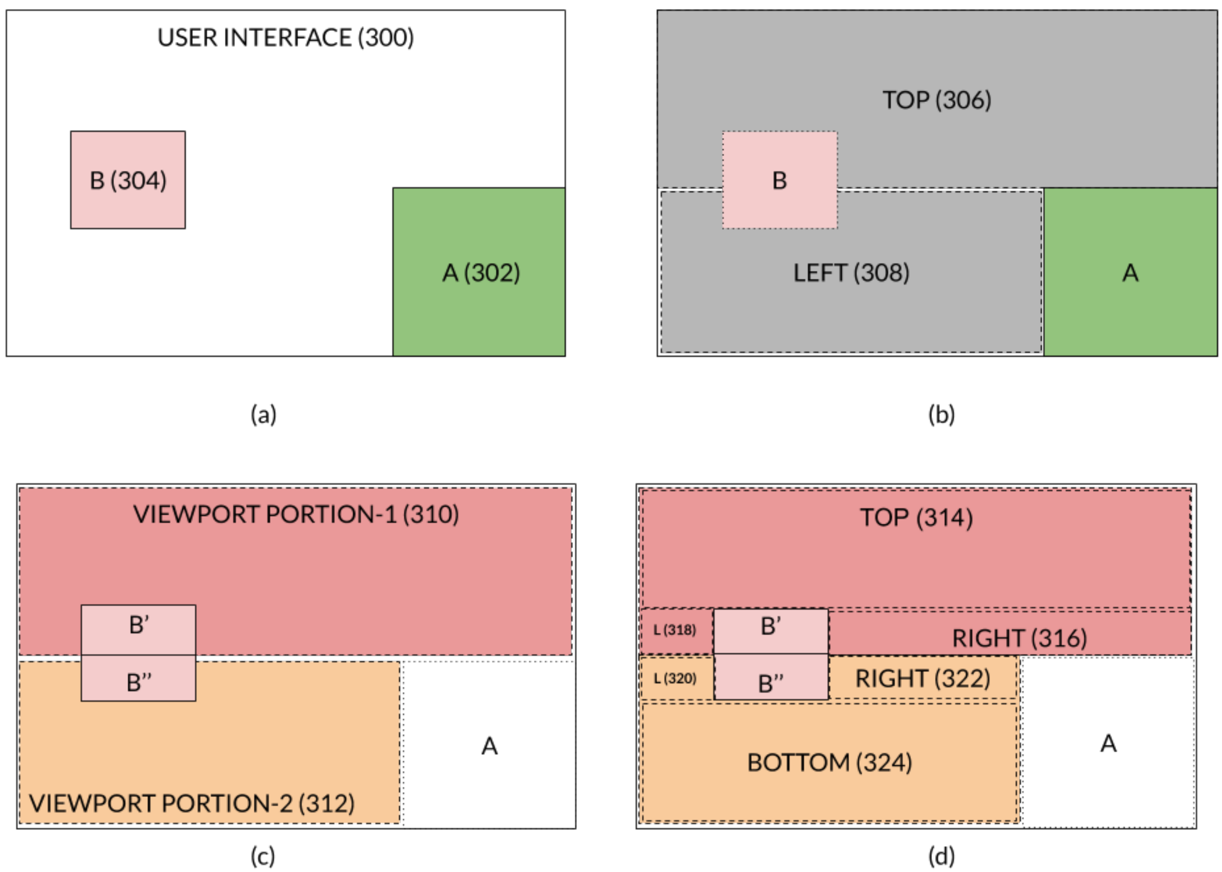

Fig. 3: Viewports represented by a set of recursively divided rectangles

Techniques of this disclosure can be applied even when the user interface includes multiple occluding rectangles. Fig. 3(a) depicts an example user interface (300), with two occluding rectangles, A (302), and B (304). In this illustrative example, the viewport is the area of the user interface that is unoccluded by rectangles A and B.

Determination of the viewport area in this case is performed in a two-step process. In a first step, as depicted in Fig. 3(b), an area of the user interface that is not occluded by rectangle A is determined by representing the unoccluded portion by a top (306) rectangle and a left (308) rectangle. In this illustrative example, bottom and right rectangles are not utilized in the representation of the user interface area unoccluded by rectangle A.

In a second step, the second occluding user interface element B is considered. As depicted in Fig. 3(c), rectangle B straddles the top and left rectangles determined in the previous step. The user interface is therefore subdivided into a viewport portion-1 (310) that is occluded by rectangle B’ and a viewport portion-2 (312) that is occluded by rectangle B’’, where rectangles B’ and B’’ together constitute occluding rectangle B.

Fig. 3(d) depicts the recursive subdivision of viewport portion-1 and viewport portion-2. The unoccluded portion of viewport-1 is represented by left (L) rectangle (318), right rectangle (316), and top rectangle (314), while the unoccluded portion of viewport-2 is represented by left (L) rectangle (320), right rectangle (322), and bottom rectangle (324).

In this manner, the area of the user interface unoccluded by rectangles A and B are efficiently represented by a set of rectangles. In this illustrative example, the viewport area is determined by the determination of the total area of the set of rectangles, e.g. total area represented by L (318), right (316), top (314), L (320), right (322), and bottom (324) rectangles.

Efficient representation of the unoccluded space (e.g. the viewport) with a small number of rectangles using recursive subdivision as described herein enables frequent recalculation of the viewport area, e.g. multiple times per second. The techniques therefore support viewport determination even when the viewport changes dynamically, e.g., as the user scrolls through content or performs other actions that modify the viewport, and user interfaces of arbitrary complexity.

CONCLUSION

This disclosure describes techniques for the determination of occluded and unoccluded spaces within a user interface. Per techniques of this disclosure, a viewport of a user interface is represented as a set of rectangles determined by recursive subdivision. The area of the viewport is based on a sum of the areas of each unoccluded rectangle in the set of rectangles. The efficient representation of the viewport with a small number of rectangles enables frequent recalculation of the viewport, e.g., even when the viewport is updated frequently in response to user actions such as scrolling.